

Revista
Española de
Innovación,
Calidad e
Ingeniería del Software



Volumen 4, No. 3, octubre, 2008

Web de la editorial: www.ati.es/reicis

E-mail: editor-reicis@ati.es

ISSN: 1885-4486

Copyright © ATI, 2008

Ninguna parte de esta publicación puede ser reproducida, almacenada, o transmitida por ningún medio (incluyendo medios electrónicos, mecánicos, fotocopias, grabaciones o cualquier otra) para su uso o difusión públicos sin permiso previo escrito de la editorial. Uso privado autorizado sin restricciones.

Publicado por la Asociación de Técnicos de Informática

www.ati.es



Revista Española de Innovación, Calidad e Ingeniería del Software (REICIS)

Editores

Dr. D. Luís Fernández Sanz

Departamento de Ciencias de la Computación, Universidad de Alcalá

Dr. D. Juan José Cuadrado-Gallego

Departamento de Ciencias de la Computación, Universidad de Alcalá

Miembros del Consejo Editorial

Dr. Dña. Idoia Alarcón

Depto. de Informática
Universidad Autónoma de Madrid

Dr. D. José Antonio Calvo-Manzano

Depto. de Leng y Sist. Inf. e Ing. Software
Universidad Politécnica de Madrid

Dra. Tanja Vos

Instituto Tecnológico de Informática
Universidad Politécnica de Valencia

D. Raynald Korchia

SOGETI

D. Rafael Fernández Calvo

ATI

Dr. D. Oscar Pastor

Depto. de Sist. Informáticos y Computación
Universidad Politécnica de Valencia

Dra. Dña. María Moreno

Depto. de Informática
Universidad de Salamanca

Dra. D. Javier Aroba

Depto de Ing.El. de Sist. Inf. y Automática
Universidad de Huelva

D. Antonio Rodríguez

Telelogic

Dr. D. Pablo Javier Tuya

Depto. de Informática
Universidad de Oviedo

Dra. Dña. Antonia Mas

Depto. de Informática
Universitat de les Illes Balears

Dr. D. José Ramón Hilera

Depto. de Ciencias de la Computación
Universidad de Alcalá

Contenidos

REICIS

Editorial	4
<i>Luís Fernández-Sanz, Juan J. Cuadrado-Gallego</i>	
Presentación	5
<i>Luis Fernández-Sanz</i>	
El tamaño sí importa en la mejora de procesos	6
<i>Tanja E.J. Vos, Jorge Sánchez Sánchez y Maximiliano Mannise</i>	
La gestión de la configuración y la gestión de activos como una gestión del conocimiento	18
<i>Jesús García Romanos</i>	
Reseña sobre las V Jornadas de Testeo de Software (JTS'08)	36
<i>Tanja E.J. Vos</i>	
Reseña sobre el III Congreso Interacadémico 2008-itSMF España	38
<i>Antonio Folgueras</i>	
Sección Actualidad Invitada:	40
Un sociólogo estudia la producción de software: ¿qué puede aportar su mirada al estudio de la evolución del trabajo de los informáticos?	
<i>Juan José Castillo Alonso, Director del Grupo de Investigación 'Charles Babbage' en Ciencias Sociales del Trabajo, Facultad de Ciencias Políticas y Sociología, Universidad Complutense de Madrid</i>	

Editorial

The logo for REICIS, consisting of the word "REICIS" in white, bold, uppercase letters on a black rectangular background.

Tradicionalmente la calidad del software se ha considerado dependiente de tres factores principales involucrados en su desarrollo: la tecnología, el proceso y las personas. Los aspectos técnicos y de proceso suelen ser exhaustivamente tratados en los distintos canales de difusión y debate relacionados con el software. Lamentablemente, al tratarse el software de un producto eminentemente intelectual, hay que relacionar En este número, algunos aspectos relacionados con este aspecto humano tan frecuentemente olvidado en la ingeniería del software son abordados en nuestra sección de contribuciones invitadas por un experto de larga trayectoria en la investigación del factor humano en la informática: D: Juan José Castillo Alonso, catedrático de sociología y director del Grupo de Investigación 'Charles Babbage' en Ciencias Sociales del Trabajo (www.ucm.es/info/charlesb/) en la Facultad de Ciencias Políticas y Sociología de la Universidad Complutense de Madrid. En su contribución comenta como el enfoque sociológico del trabajo en el desarrollo de software puede aportar un mejor conocimiento de la estructura del trabajo en los proyectos y en las organizaciones, peculiarmente con fenómenos nuevos como las factorías de software.

Desde REICIS queremos contribuir a que el aspecto humano en la ingeniería del software sea fuente de inspiración cada día más intensa para los investigadores y profesionales en el ámbito de la innovación y la calidad del software. En este sentido, esperamos que sean muchos y variados los autores que acepten este reto y que deseen difundir sus avances utilizando REICIS como el medio más apropiado para dar a conocer sus trabajos e investigaciones teniendo las máximas garantías de la profesionalidad con que serán tratados sus trabajos y aprovechando su cada vez más amplia difusión en todo el mundo de habla hispana. Podrán encontrar todas las instrucciones necesarias para el envío de sus contribuciones en la página web de la revista: www.ati.es/reicis.

Luis Fernández Sanz
Juan J. Cuadrado-Gallego
Editores

Este tercer número de REICIS del año 2008 (tras incorporar el número especial dedicado a las X Jornadas de Innovación y Calidad del Software) publica la sección regular de artículos con dos contribuciones que son fruto del acuerdo con distintos eventos de interés y prestigio en España. Por una parte, se cuenta con el acuerdo con la V Edición de las Jornadas de Testeo de Software (JTS'08), organizadas por el Instituto Tecnológico de Informática de la Universidad Politécnica de Valencia y celebradas los días 2 a 4 de abril de 2008. El proceso, como ya es conocido, incluye la selección de los originales más prometedores para posteriormente solicitar a sus autores una versión mejorada y extendida. Esta versión es finalmente revisada por el comité editorial solicitando las mejoras necesarias para su publicación en REICIS. Finalmente se ha seleccionado el trabajo denominado “El tamaño sí importa en la mejora de procesos” realizado por T. Vos, J. Sánchez y M. Maníes. Para completar la cobertura de esta reunión científica, este número incluye una reseña sobre las JTS'08 a cargo de su responsable, Tanja Vos.

Por otra parte, también se publica un trabajo seleccionado de entre los presentados al III Congreso Interacadémico ITSME-España que se celebró a lo largo del día 13 de octubre de 2008 en la Universidad Carlos III de Madrid. Se trata de la contribución titulada “La gestión de la configuración y la gestión de activos como una gestión del conocimiento” a cargo de Jesús García Romanos de la empresa IBM donde se analiza la problemática de la gestión de la configuración y de otros activos dentro del marco de ITIL v3. Por supuesto, también se incluye una reseña sobre el congreso a cargo de Antonio Folgueras, de la Universidad Carlos III de Madrid y responsable del mismo.

Finalmente, como ya hemos comentado en el editorial, incluimos en la columna de Actualidad Invitada, la contribución titulada “Un sociólogo estudia la producción de software: ¿qué puede aportar su mirada al estudio de la evolución del trabajo de los informáticos?” de Juan José Castillo Alonso sobre el estudio del trabajo y del factor humano en el desarrollo de software.

Luis Fernández Sanz

El tamaño sí importa en la mejora de procesos

Tanja E.J. Vos, Jorge Sánchez Sánchez, Maximiliano Mannise
SQUaC (*Software Quality, Usability and Certification*)
Instituto Tecnológico de Informática
Universidad Politécnica de Valencia,
Camino de Vera s/n - 46022 Valencia, Spain
{tanja,jordisan,mmannise}@iti.upv.es

Resumen: A partir de las experiencias en la aplicación de procesos de mejora de testeo en diferentes PYME, detectamos que las metodologías más habituales de ese tipo están orientadas a organizaciones mucho más grandes y son, por tanto, difícilmente aplicables a ese perfil de empresas, con recursos limitados y poca madurez en los procesos de testeo. Proponemos, por tanto, una serie de acciones sencillas y concretas que las PYME pueden realizar sin dedicar muchos recursos, obteniendo resultados rápidamente y preparándolas para un posterior proceso de mejora más formal.

Palabras clave: Testeo, software, PYME, mejora, procesos

Abstract: Based on experiences in implementing testing processes improvement in different SMEs, we found that the most common methodologies are aimed at much larger organizations and are, therefore, difficult to implement in a company profile that has limited resources and low maturity in the testing processes. We therefore propose a series of simple and concrete actions that SMEs can do to spend little resources and get fast results, while getting prepared for further, more formal improvement processes.

Keywords: software, testing, SME, process, improvement

1. Introducción

Este artículo describe nuestras experiencias durante los proyectos de transferencia de tecnología y servicio de consultoría con objeto de ayudar a las empresas a resolver un “problema con el testeo” detectado por ellas, o bien únicamente para apoyarlas en la mejora de sus prácticas en testeo de software. En los siguientes apartados describiremos el entorno y las características de dichos proyectos.

1.1. El ITI (Instituto Tecnológico de Informática)

El ITI es un instituto de investigación sin ánimo de lucro situado en las instalaciones de la Universidad Politécnica de Valencia (UPV), cuyo principal objetivo es aplicar los

conocimientos obtenidos de la investigación e innovación científica en tecnología punta a las empresa de pequeño y mediano tamaño (PYME) relacionadas con tecnologías de información y comunicación (TIC).

El instituto consta de siete grupos de investigación, muchos de ellos liderados por profesores universitarios. Las experiencias descritas en este artículo provienen del grupo SQUaC, que trabaja en las áreas de calidad, usabilidad y certificación de software.

La misión del ITI incluye, como uno de sus principales objetivos, construir puentes entre el conocimiento y la tecnología obtenidos de la investigación científica, por un lado, y la necesidad de soluciones simples, prácticas y robustas para la industria relacionada con las TIC, por otro. Para conseguir este objetivo, el ITI apoya a las empresas con formación a medida, transferencia de tecnología, soluciones estándar propias, y proyectos de consultoría en la propia empresa.

1.2. El ITI (Instituto Tecnológico de Informática)

Muchas de las empresas a las que apoyamos son pequeñas compañías de software en Valencia, España. Estas empresas se caracterizan por lo siguiente:

- Venden un producto software personalizado para determinados usuarios finales.
- El software, inicialmente programado únicamente por uno o dos desarrolladores, ha crecido desde algo pequeño, transparente y fácil de mantener hasta ser un monstruo enorme, opaco e imposible de mantener por un mayor grupo de desarrolladores.
- Sus clientes están malacostumbrados; primero, porque se les permite proporcionar requerimientos muy vagos para los nuevos desarrollos. Segundo, porque están acostumbrados a que todas sus demandas sean implementadas. Esto lleva frecuentemente a que existan múltiples versiones del mismo software.
- No existen documentos de requerimientos escritos, o los que existen están poco estructurados, son poco detallados o incompletos.
- No existen procesos de testeo estructurados ni personal específicamente dedicado al testeo. De hecho, en muchas ocasiones ni siquiera se realizan actividades específicas de testeo.

- Existe escasa información y gestión de los defectos en el software. La que existe está muy centrada en las incidencias notificadas por clientes.

1.3. Sus proyectos (y sus problemas)

En muchas ocasiones las empresas acuden a nosotros debido a que tiene problemas que creen que pueden resolverse mediante testeos. Los problemas son muchas veces los mismos:

- Los clientes se quejan porque encuentran muchos errores, o porque las funcionalidades de las aplicaciones no se corresponden con sus necesidades.
- No se cumplen los plazos de entrega y presupuesto acordados. En muchos casos se sacrifica el testeo de software por urgencias del proyecto.
- No se gestionan adecuadamente los defectos encontrados durante las pruebas internas, ni tampoco los reportados por los clientes.
- Se dedican demasiadas horas a corregir defectos y/o malentendidos en los requerimientos.
- Las múltiples versiones del software son difíciles de gestionar y/o testear.
- Existe poca o ninguna información sobre la calidad de los desarrollos.

1.4. El testeo de software y sus niveles

Para cualquier compañía (también para las PYME) que se dedique al desarrollo de software, sea cual sea la metodología que utilice, el testeo debe ser una parte fundamental tanto para asegurarse de que los productos están siendo desarrollados correctamente (verificación) como de que los productos satisfarán los requisitos del usuario (validación).

En cualquier caso, un testeo adecuado durante el ciclo de desarrollo cumple que ([1]):

- El testeo se inicia en las primeras etapas del ciclo de vida del software
- No sólo se testea el software final, sino también los requerimientos, manuales, otros documentos, etc.
- Los testadores están involucrados en la captura de requerimientos

Los niveles típicos de testeo durante el ciclo de vida de una aplicación son:

- Testeo unitario (o de componentes). El código fuente del software suele estar dividido en unidades más o menos aisladas, también llamadas programas, módulos, componentes o clases. El testeo unitario pretende asegurar que cada una

de esas unidades cumple su especificación por separado. Este testeo es habitualmente realizado por el mismo desarrollador que escribe el código.

- Testeo de integración. Una vez que las unidades han sido escritas y testeadas, la siguiente etapa consiste en unir las para crear el sistema completo; esto es llamado integración. El propósito del testeo de integración es descubrir defectos en las interfaces y en las interacciones entre diferentes componentes o sistemas.
- Testeo de sistema. Una vez que se ha comprobado que los componentes funcionan y se integran correctamente entre ellos, el siguiente paso es considerar la funcionalidad completa del sistema como un todo; a esto se le llama testeo de sistema. Este testeo es necesario ya que los niveles inferiores de testeo (unitario y de integración) utilizan una visión parcial y no son representativos de las condiciones reales de funcionamiento del sistema. El testeo de sistema, usualmente realizado por un equipo ajeno a su desarrollo, puede referirse tanto a requerimientos funcionales (comprobando que el sistema realiza las funciones deseadas) como a requerimientos no funcionales, como son: rendimiento, carga, fiabilidad, usabilidad, etc.
- Testeo de aceptación. El propósito del testeo de aceptación es dar confianza al usuario final de que el sistema funcionará de acuerdo a sus expectativas. La base para el testeo de aceptación es el documento de especificación de requerimientos, comprobando en la práctica que el sistema los cumple adecuadamente. Este testeo suele ser totalmente independiente del resto de niveles de testeo y de la implementación del sistema, y puede ser llevado a cabo por los propios usuarios finales del sistema (con la posible participación de miembros del equipo de desarrollo), en las instalaciones de desarrollo o en las del propio cliente.

2. La mejora de procesos y las PYME

2.1. Metodologías de mejora de procesos

Existen bastantes metodologías y técnicas para la mejora de procesos en general, y para la mejora de procesos de testeo en particular (algunas de ellas mencionadas en [2] y [3]).

Metodologías como SPICE [4], desarrollada por ISO/IEC, pretenden ser un estándar para la evaluación de procesos software en general. Otras, como CMMI [5], se orientan de manera todavía más global, abarcando todo el desarrollo y mantenimiento del producto, e integrando diferentes áreas de conocimiento que antes eran tratadas por separado, como ingeniería de software, ingeniería de sistemas o aprovisionamiento.

Otros autores, ante la insuficiente atención que prestan esos modelos globales al testeo de software, han desarrollado metodologías de mejora específicas como Test Organization Maturity (<http://www.gerrardconsulting.com/default.asp?page=/tomoverview.html>), Test Improvement Model (TIM) [6], Testing Maturity Model (TMM) [7] o Test Process Improvement (TPI) [8].

2.2. Mejora de procesos en PYME, en la práctica

El mayor problema con los modelos anteriores es que fueron diseñados inicialmente para organizaciones más grandes, y no son fácilmente adaptables a estructuras mucho más pequeñas, como nuestras PYME. El primer motivo es que el coste y la duración de un proceso de evaluación son desproporcionados respecto a los recursos disponibles en una PYME. Además, su nivel de madurez, especialmente en lo que respecta a testeo de software, es normalmente muy bajo; por ejemplo, en una evaluación inicial rápida, la mayoría de las PYME con las que trabajamos únicamente alcanzan el nivel mínimo de madurez de TPI (el nivel A) en el área de entorno de oficina.

Varios estudios (por ejemplo [10]) muestran que un gran número de proyectos de mejora de procesos en PYME basados en modelos como CMMI encuentran problemas graves, y que un importante porcentaje de esos problemas (53%) está relacionado con el tamaño de la empresa; el éxito de un proyecto de mejora de procesos crece con el número de personas a cargo de procesos de software. Como se indica en [11], en este tipo de proyectos y con empresas de ese tamaño, es imposible recoger suficiente material como para realizar un análisis estadístico sólido, siendo en estos casos mucho más importante el factor personal, involucrando desde el principio a las personas clave (las que poseen los conocimientos).

Por otro lado ([12]), la rigidez y el formalismo de esas metodologías conllevan para las PYME un incremento de la burocracia que se convierte en un lastre ante la flexibilidad y la innovación que dichas empresas necesitan en un entorno dinámico sobre el que

prácticamente no tienen control. Reducir el rigor de las metodologías no basta para reducir esos inconvenientes.

Un factor clave en el éxito o fracaso de este tipo de proyectos reside en la motivación que impulsa a la PYME a emprender ese proceso de mejora ([13]). Para algunas se trata de un “mal necesario” que deben afrontar por la presión de clientes importantes como organismos públicos o grandes empresas; en estos casos el proyecto se considera simplemente como un coste más, sin auténticas expectativas de mejora. Por otro lado, cuando los responsables de la empresa ven en el proyecto de mejora una verdadera oportunidad de optimizar en la práctica sus procesos y sistemas internos, aumentan las posibilidades de obtener resultados positivos.

En el caso concreto de España, estudios como [9], además de resaltar la especial importancia de las PYME (representando más de un 99% de las empresas en el sector software), ponen de manifiesto que metodologías como CMMI y SPICE son bastante conocidas, gracias en gran medida a las subvenciones de organismos públicos para su adopción. Sin embargo, parece que el interés de las empresas está más centrado en la certificación frente a terceros que en la mejora interna de procesos: existe conocimiento sobre la existencia de los estándares y su utilización fuera de España, pero no se conocen los detalles sobre su uso interno y todo aquello que implica su adopción a la hora de aplicarlas en la práctica. Como resultado, el nivel de adopción es muy bajo, siendo las barreras que esgrimen las PYME los altos costes de la adopción del modelo, el cambio en los métodos de trabajo que implican, la necesidad de realizar un cambio organizativo o la dificultad de evaluar los beneficios de implantarlas.

En resumen, las experiencias de las PYME con las metodologías de mejora de procesos son muchas veces problemáticas, debido a la inversión excesiva en tiempo, en herramientas, etc., que les supone aplicar modelos que están pensados para organizaciones más grandes, con más recursos y mayor madurez en esos procedimientos. Esas dificultades se convierten en un abismo que separa a la PYME de los procesos de mejora (Figura 1)..

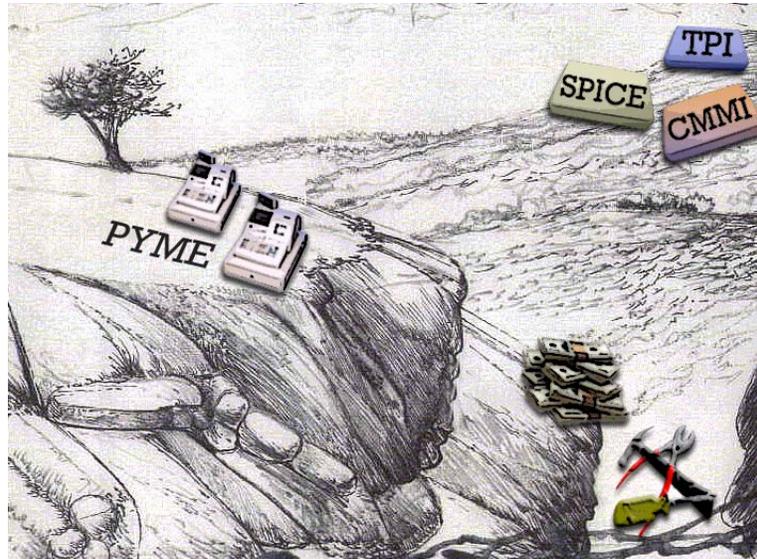


Figura 1. En el abismo entre las PYME y las metodologías de mejora se pierden muchos recursos: dinero gastado, herramientas abandonadas, etc..

3. Preparando los modelos de mejora para las PYME

Ante todos esos problemas que surgen a la hora de aplicar los modelos de mejora a las PYME, se han realizado diversos intentos de adaptar o crear nuevos modelos específicamente orientados a las características de empresas de pequeño tamaño.

En general, consisten en simplificaciones y/o adaptaciones de modelos más complejos como CMMI (que es usado como referencia en IT Mark¹, especialmente implantada en España) o SPICE/ISO 15504 (que es tomado como base de Rapid Assessment for Process Improvement for software Development, RAPID [15]); o en la combinación de varios de ellos (como hace [16]).

Otros modelos, como Minimal Test Practice Framework (MTPF) [17], están específicamente centrados en las actividades de testeo de software e intentan adaptarse a la naturaleza concreta de las PYME: escasez de recursos, resistencia al cambio interno, flexibilidad, etc.

En España según [9], ITmark parece tener especial importancia, aunque existen también iniciativas como el programa SoftAragón, que ha desarrollado un modelo CMMI

¹ <http://www.esi.es/index.php?op=15.1.2>

reducido para su aplicación a las PYME aragonesas, o MESOPYME [18], especialmente orientado a la fase de implementación de la mejoras.

Todas estas adaptaciones hacen que los modelos de mejora (y, en concreto, los de mejora del testeado) sean más fácilmente aplicables a las PYME; pero esas metodologías específicas pueden no ser suficientes, sobre todo debido a la poca madurez y conciencia que encontramos en la práctica en las empresas. ¿No hay nada que podamos hacer en la propia PYME para facilitar la implementación de una metodología de mejora de procesos de testeado? Nosotros creemos que sí.

4. Preparando a las PYME para la mejora de procesos de testeado

Antes de empezar a aplicar alguno de los modelos de mejora vistos anteriormente (tanto generales como adaptados a pequeñas organizaciones) en una PYME con las características que veíamos al principio (ausencia de procesos de testeado estructurados, sin personal dedicado, etc.), proponemos preparar a la empresa con dos acciones muy prácticas y sencillas:

- Testeo unitario por programadores
- Personal exclusivamente dedicado al testeado

4.1 Testeo unitario por programadores

La única responsabilidad de los programadores en cuanto al testeado sería realizar algún tipo de testeado unitario sobre sus desarrollos, dejando a su elección qué y cómo hacerlo. Confiemos en el conocimiento de los programadores: saben lo que quieren programar, así que dejemos que sean ellos los que comprueben si lo hacen correctamente.

Para tener éxito en este punto sería necesario:

- Obligar a los programadores a que realicen testeado unitario. Al fin y al cabo, la tarea de los programadores no es simplemente producir programas, sino producir código de calidad, libre de errores; y una de las maneras de conseguirlo es mediante el testeado.
- Motivar a los programadores para que realicen el testeado. Para ello pueden utilizarse diferentes técnicas. Por ejemplo, hacer reuniones periódicas específicamente dedicadas al testeado (“cada viernes por la mañana”); realizar seminarios informales

para exponer los avances conseguidos (“mirad lo que he conseguido/encontrado esta semana”); o utilizar otros medios para compartir esa información, como Intranets [12] o incluso el “testeo en el aseo” que realiza Google (<http://googletesting.blogspot.com/2007/01/introducing-testing-on-toilet.html>).

- Darles libertad para realizar el testeo según crean conveniente. De hecho, los programadores se enfrentan continuamente con nuevas tecnologías y técnicas que tienen que utilizar para resolver algún tipo de problema; constantemente buscan, aprenden y aplican esas nuevas tecnologías. Así que ¿por qué no van a hacerlo con los frameworks de testeo unitario y otras herramientas existentes?
- Proporcionarles los recursos necesarios. Sobre todo necesitarán tiempo asignado específicamente al testeo, tanto para aprender e investigar los mejores métodos como para realizar efectivamente el testeo unitario. Por su trabajo, normalmente ya dispondrán del entorno (hardware, instalaciones, etc.) requerido para el testeo, pero si no se planifica el esfuerzo extra de realizar el testeo unitario, lo más probable es que no le dediquen el tiempo necesario.

Para complementar sus conocimientos, podría ser necesario algún tipo de formación orientada a explicar en términos generales qué es testeo unitario, así como una visión global de las técnicas y herramientas existentes. Todo ello sin entrar en detalles, dejando que sean los programadores los que elijan cómo aplicar en la práctica esos conocimientos a su entorno de trabajo.

4.2. Personal exclusivamente dedicado al testeo

Consistiría en contratar al menos 1 persona dedicada exclusivamente al testeo de alto nivel. Asignándole como único objetivo el encontrar errores, además de encontrarlos en la práctica, sacará a la luz necesidades para realizar correctamente su trabajo (como mejor definición de requerimientos, gestión de defectos, análisis de riesgos, planificación, etc.) que, además mejorarán los procesos de desarrollo de la empresa.

Idealmente, el responsable de esa tarea sería alguien con conocimientos y experiencia previa en testeo de software. No obstante, teniendo en cuenta el mercado laboral actual y la dificultad de encontrar y contratar profesionales con esos requisitos (especialmente para las PYME), podemos aceptar que se trate de una persona sin experiencia previa en testeo. En general, se trataría de alguien con perfil técnico (por

ejemplo, Ingeniero en Informática) y ciertas habilidades en otros campos: comunicación, planificación de tareas, etc.; en esta situación será muy importante su capacidad de autoformación. En todo caso, recordemos que estas sugerencias pretenden conducir, no a la estructura de testeo definitiva dentro la organización, sino a una preparación básica previa antes de acometer un proceso de mejora más formal.

La persona asignada a esa tarea podría ser alguien que ya pertenece a la empresa, aunque en la práctica suele ser muy difícil que una persona que ya tiene responsabilidades asignadas abandone todas sus tareas anteriores para dedicarse únicamente al testeo. Contratando nuevo personal se evita ese problema, lo que compensa otros inconvenientes, como el aumento de costes o el desconocimiento que alguien venido de fuera pueda tener de la organización.

4.3. Otros aspectos

Además de esos dos puntos, pueden ser de utilidad algunos cursos básicos sobre testeo para incrementar la conciencia de su necesidad en la empresa, aunque no demasiados para que la PYME no se vea abrumada por la cantidad de cosas que NO están haciendo.

Tenemos que estar preparados para que la PYME sea en ocasiones reacia a aceptar esa libertad (“que hagan lo que crean necesario”) al principio. Sin embargo, es fácil explicarles que, si partimos de una situación en la que prácticamente nada se está haciendo, cualquier paso es ya un gran avance en el camino hacia la mejora de procesos. Es importante que la PYME entienda que, en este primer momento, el factor más importante es la motivación.

Así, tras algún tiempo (proponemos unos 6 meses), la empresa habrá adquirido ciertos conocimientos y experiencias en testeo, y podremos empezar con acciones más ambiciosas: aplicar una metodología más formal como TPI, cursos más específicos en función de las necesidades de la PYME, etc.

5. Conclusiones

Con las dos sugerencias que ofrecemos, y basándonos en nuestra experiencia, creemos que una PYME se encontrará en poco tiempo en una situación mucho más propicia para poder acometer con éxito un proceso más ambicioso de mejora del testeo, como pueda ser aplicar

una metodología de mejora más formal, aumentar los recursos dedicados al testeo, etc. En todo caso, suponen una inversión relativamente pequeña y cualquier avance en la madurez de la empresa en temas de testeo que se consiga con estas prácticas supondrá un avance importante respecto a la situación de partida que nos encontramos habitualmente.

Algunas de las PYME con las que hemos trabajado están empezando a aplicar (de modo parcial) estas sencillas sugerencias, obteniendo de inicio resultados bastante positivos. Nos queda pendiente estudiar y reflejar las experiencias y los resultados obtenidos en la práctica después de transcurrido un tiempo suficiente.

Referencias

- [1] Hambling, B., Morgan, P., Samaroo, A., Thompson, G. y Williams, P., *Software Testing. An ISEB Foundation*. The British Computer Society, 2007.
- [2] Koomen, T. y Pol, M., *Test Process Improvement: A practical step-by-step guide to structured testing*, Addison-Wesley, 1999.
- [3] Swinkels, R., *Technical Report 12-4-1-FP. A Comparison of TMM and Other Test Process Improvement Models*, Frits Philips Institute, 2000.
- [4] Kulkarni, S., “Test Process Maturity Models – Yesterday, Today and Tomorrow”. *Proceedings of the 6th Annual International Software Testing Conference, Delhi, India*, marzo, 2006.
- [5] Chrissis, M. B., Konrad, M. y Shrum, S., *CMMI: Guidelines for Process Integration and Product Improvement*. Addison-Wesley, 2003.
- [6] El Emam, K., Drouin, J.N. y Melo, W., *SPICE: The Theory and Practice of Software Process Improvement and Capability Determination*. Wiley-IEEE Computer Society Press, 1997.
- [7] Ericson, T., Subotic, A. y Ursing, S., “TIM-a test improvement model. Software Testing”, *Verification and Reliability*, vol. 7, num. 4, pp. 229-246, 1997
- [8] Burnstein, I., Suwannasart, T. y Carlson, C. "Developing a Testing Maturity Model". *Crosstalk: The Journal of Defense Software Engineering*. Part I, vol. 9, num. 8, pp. 21-24, Part II, vol. 9, num. 9, pp. 19-26, 1996.

- [9] INTECO, *Estudio sobre la certificación de la calidad como medio para impulsar la industria de desarrollo del software en España*. Instituto Nacional de Tecnologías de la Comunicación, abril 2008.
- [10] Brodman, J. G. y Johnson, D. L., “What Small Businesses and Small Organisations say about CMM?”, *Proceedings of the 16th International Conference on Software Engineering, Sorrento, Italia, mayo 1994*, pp. 331 – 340, 1994.
- [11] Lied, H.J. y Stlhane, T., “Experience from process improvement in a SME”, *Proceedings of the EuroSPI'99 Conference Pori, Finland, 25-27 octubre, 1999*.
- [12] McAdam, R., “Quality models in an SME context: A critical perspective using a grounded approach”. *International Journal of Quality & Reliability Management*, vol.17, num. 3, pp. 305-323, 2000.
- [13] Brown, A., Wiele, T.V.D. y Loughton, K., “Smaller enterprises’ experiences with ISO 9000”, *International Journal of Quality & Reliability Management*, vol. 15, num. 3, pp. 273-285, 1998.
- [14] Rout, T., Tuffley, A, Cahill, B y Hodgen, B., “The Rapid Assessment of Software Process Capability”, *Proceedings of SPICE 2000, Limerick*, pp. 47-55, 2000.
- [15] Alexandre, S., Renault, A. y Habra, N., “OWPL Software Process Improvement for VSE, SME and low maturity enterprises”, *Conference on Software Engineering and Advanced Applications, 2006. SEAA '06. 32nd EUROMICRO*, pp. 328-335, 2006.
- [16] Karlström, D., Runeson, P. y Norden, S., “A Minimal Test Practice Framework for Emerging Software Organisations. Software Testing”, *Verification and Reliability*, vol.15, num. 3, pp. 145-166, 2005.
- [17] Calvo-Manzano, J. A., Cuevas, G., San Feliu, T., De Amescua, A., García, L. y Pérez, M., “Experiences in the Application of Software Process Improvement in SMES”, *Software Quality Control*, vol. 10, num. 3, pp. 261-273, 2002.

La gestión de la configuración y la gestión de activos como una gestión del conocimiento

Jesús García Romanos
IBM Tivoli Business Automation
IBM España
JGRomanos@es.ibm.com

Abstract

The effectiveness and efficiency of organizations depends on the correct handling of its assets, especially those that are used to provide services to customers or those that are vital to the smooth running of the business. ITIL v3 expressly recognises the value of assets for service management, expanding the scope of configuration management and the CMDB tools which support service management processes. The paper describes the scope of asset management and configuration management, both closely linked in, and being part of a comprehensive knowledge management. We describe an approach to draw up an asset and configuration management process based on the automation of the various stages of these processes, which reduces the risks of their implementation. The steps to automate include the auto discovery of assets and configuration elements, and the synchronization and reconciliation of the different sources of data, both key steps previous to provide support to the other various service management processes.

Key words: Asset, ITIL (IT Infrastructure Library), CMDB (Configuration Management Database), Service Management, SKMS (Service Knowledge Management System).

Resumen

La eficacia y eficiencia de las organizaciones depende del correcto manejo de sus activos, en especial de aquellos que sirven para prestar servicio a sus clientes o aquellos que son vitales para el funcionamiento del negocio. ITIL V3 reconoce expresamente el valor de los activos para la gestión del servicio TI, expandiendo el alcance de la gestión de la configuración y de las herramientas de CMDB en las que se apoyan los procesos de gestión del servicio. En el documento se describe el alcance de la gestión de la configuración y el de la gestión de activos, ambos estrechamente ligados entre si, formando parte de una gestión integral del conocimiento. Abordamos una aproximación para elaborar una gestión de activos y de configuración integrada mediante la automatización de las distintas etapas de estos procesos, lo que reduce los riesgos de su implementación. Entre las etapas a automatizar destacamos el autodescubrimiento de los activos y de los elementos de configuración, y la sincronización y reconciliación de las distintas fuentes de datos como paso previo a dar soporte a los distintos procesos de gestión del servicio.

Palabras clave: Activos, ITIL, Base de Datos de configuración (CMDB), Gestión del Servicio, Sistema de Gestión del Conocimiento del Servicio (SKMS).

1. Introducción

Para gestionar los servicios que se prestan desde los departamentos de tecnologías de la información (TI), es necesario conocer los activos disponibles para proporcionar estos servicios, y conocer cómo se configuran e interrelacionan los distintos elementos para prestar los servicios. Mantener la información actualizada de los activos y de los elementos de configuración es la función de la Gestión de la Configuración y Activos del Servicio. En este documento describimos como abordar esta gestión de manera eficiente.

En primer lugar se describen las mejores prácticas de la industria, empezando por una descripción de ITIL y de las normas ISO2000 asociadas, para luego presentar otros marcos y modelos de referencia como CBMBoIT, CMMI, y COBIT. En el siguiente apartado se hace un análisis de las diferencias entre elemento de configuración, activo y conocimiento en el marco de ITIL V3 para luego describir el proceso de Gestión de la Configuración y Activos de Servicio.

Por último se aborda una aproximación para implementar una gestión de activos y configuración que siguiendo las prácticas reconocidas de TI, sea abordable con los menores riesgos posibles en base a la automatización de las distintas etapas del proceso de la gestión de activos y configuración.

2. Las mejores practicas de la industria

ITIL es una recopilación de mejores prácticas de las Tecnologías de la Información, internacionalmente reconocida y en constante evolución, diseñada para ayudar a las organizaciones a superar los retos actuales y futuros. Creada originalmente por la administración del Reino Unido en 1988, es el resultado de años de experiencia a los que han contribuido las principales organizaciones y compañías de TI, incluida IBM. Los departamentos de TI de todo el mundo utilizan ITIL como base para guiarse en la implementación eficaz de una estrategia de gestión de servicios de TI. ITIL es propiedad de la *Office of Government Commerce* (<http://www.ogc.gov.uk>) del Reino Unido.

IBM ha estado relacionado con ITIL desde su inicio, y contribuyó de forma importante en la plataforma original de la biblioteca. La arquitectura ISMA (*Information Systems Management Architecture*) de IBM, desarrollada en los años 1970, y publicada por primera vez en 1980 [1] sirvió como base de muchas de las definiciones de proceso de

ITIL. Como miembro global del *IT Service Management Forum* (itSMF), la única organización internacionalmente reconocida e independiente dedicada a la gestión de servicios de TI, IBM sigue dando soporte a la creación de nuevos materiales de la biblioteca, incluida la definición de ITIL, versión 3.

ITIL y las normas asociadas ISO/IEC 20000 están ampliamente extendidas, pero existen otros modelos alternativos que complementan la mejora de procesos de TI. Cabe destacar el “Component Business Model™ for the Business of IT” (CBMBoIT) [2] que describe un modelo de gestión TI desde la perspectiva de la dirección de sistemas de información. En CBMBoIT, se definen las actividades específicas para cada componente del modelo, permitiendo la descomposición de los procesos hasta el nivel de actividad. Estas actividades agrupadas en procesos forman un modelo de referencia completo denominado “IBM Process Reference Model for IT” (PRM-IT)². PRM-IT abarca todas las actividades de TI, extendiendo el conjunto de actividades contempladas ITIL. Existe una estrecha correlación en aquellas actividades descritas tanto en ITIL como PRM-IT, pero este último modelo se extiende a todas las actividades del área de responsabilidad de la dirección de las tecnologías de la información. Otro marco de mejora de procesos utilizado ampliamente es *Capability Maturity Model Integration* (CMMI) (www.sei.cmu.edu/cmmi/) modelo de evaluación de los procesos de una organización, proporcionando un modelo de madurez de procesos en el desarrollo de software.

Otra perspectiva es la que proporciona “Control Objectives for Information and related Technology” (COBIT) (www.isca.org). COBIT identifica cuáles deberían ser los objetivos de los procesos de TI, pero no cómo alcanzar esos objetivos. Es una infraestructura orientada a la empresa que identifica objetivos de control de alto nivel, agrupados en dominios. Cada objetivo de control de alto nivel admite varios objetivos de control detallados. Cada uno de los objetivos de control hace referencia a recursos de TI, así como a los requisitos de calidad, fiduciarios y de seguridad de la información.

Todos estos marcos y mejores prácticas se encuentran descritos en la herramienta de descarga gratuita “IBM Tivoli Unified Process” (ITUP) [3] que proporciona documentación detallada de los procesos de Gestión de servicios de TI, basados en PRM-IT e ITIL.

² www306.ibm.com/software/tivoli/governance/servicemanagement/welcome/process_reference.html

Permite comprender fácilmente los procesos, las relaciones entre ellos y los roles y herramientas que participan en una implementación de procesos eficiente.

ITUP contiene diagramas y descripciones de proceso detallados que permiten comprender los procesos y sus relaciones, facilitando la implementación de las recomendaciones de las prácticas de ITIL. En ITUP se correlacionan los distintos modelos de proceso reconocidos en la industria. ITUP es una herramienta de descarga gratuita, desarrollada conjuntamente por IBM Global Services y Tivoli, que incluye guías de herramientas, roles, productos de trabajo y casos prácticos para la gestión de servicio TI



Figura 1. Estructura de IT UP

Las guías de herramientas describen mejores prácticas para utilizar herramientas TI en procesos específicos según el contexto. Una guía de herramientas ayuda a identificar qué productos y soluciones pueden utilizarse para ejecutar actividades específicas y describe de forma detallada cómo utilizar estas herramientas de forma adecuada.

El personal de TI es responsable normalmente de uno o más roles dentro de su responsabilidad en el trabajo. Estos roles están asociados a la ejecución de tareas específicas. ITUP describe estos roles y responsabilidades con detalle y proporciona orientación para permitir al personal realizar su trabajo de forma más efectiva y eficiente.

Los productos de trabajo, definidos formalmente y sujetos al control de versiones se denominan artefactos. Las entradas o salidas de los procesos son artefactos producidos, utilizados o modificados en las actividades de los procesos. ITUP describe los productos de trabajo de cada proceso e información adicional como las definiciones de términos clave.

Los casos prácticos que se encuentran en ITUP describen problemas habituales y la solución recomendada. Con estos casos de ejemplo, podemos ver cómo afrontar los problemas del mundo real mejorando e integrando procesos, utilizando correctamente la herramienta adecuada y configurando los roles y responsabilidades necesarios.

3. Configuración, activos, conocimiento

Cuando analizamos la nueva versión de ITIL, la versión 3 publicada en mayo de 2007, e identificamos las novedades que aporta al conocimiento de las buenas prácticas de gestión de servicio TI, podemos afirmar que no aporta nada realmente nuevo. Sin embargo es una versión profundamente actualizada y fresca que saca a la luz muchas de los conceptos y procesos que se encontraban sobreentendidos en las versiones anteriores. ITIL V3 sigue siendo una guía de las mejores practicas en la gestión del servicio TI que ayuda a pensar en como implementar los procesos y funciones que soportan los servicios que se prestan al negocio día a día. En la nueva versión se hacen explicitas muchas actividades y tareas que se encontraban implícitas en las versiones anteriores. Entre los conceptos que se explicitan nos encontramos los conceptos de activo y de conocimiento, conceptos que se definen expresamente en ITIL V3.

En ITIL V2 [4], cuando se definen los objetivos de la gestión de configuración se indica que para una gestión efectiva y eficiente es necesario controlar la infraestructura y los servicios. Para ello la gestión de la configuración proporciona un modelo de la infraestructura o del servicio que permite la identificación, control, mantenimiento y verificación de los elementos de configuración (*Configuration Items*: CIs). El objetivo de ésta es: a) contabilizar los activos y configuraciones de la organización y sus servicios, b) proporcionar información fiable sobre las configuraciones para soportar los demás procesos de gestión, en particular los procesos de gestión de incidencias, problemas, cambios y “*releases*” y c) verificar los registros de configuración contra la infraestructura real corrigiendo las excepciones.

La automatización del proceso de gestión de la configuración pasa por apoyarse en una base de datos de configuración (CMDB) cuya principal característica, además de almacenar información sobre los distintos elementos de configuración, es la capacidad de almacenar información sobre las relaciones entre los distintos CIs y sobre los artefactos de

procesos relacionados con estos: Incidencias, Problemas, Solicitudes de Cambio, Acuerdos de Nivel de servicio etc.

El ámbito de los elementos incluidos en la CMDB y el detalle de información registrada sobre cada elemento depende de cada organización, pero debe adaptarse, en todo caso, a la información requerida por los distintos procesos a los que la CMDB da soporte. La CMDB puede integrar la biblioteca de software definitivo (DSL), la biblioteca de documentación e información de usuarios, personal y unidades de negocio. Por lo visto en las implementaciones de nuestros clientes, es común que esta información adicional resida en sus propios repositorios y se integre con la CMDB mediante referencias cruzadas.

Esta práctica común queda recogida en ITIL V3. La gestión de configuración en ITIL V2 pasa a Gestión de Activos de Servicio y Configuración en la Versión 3, incluyéndose en el libro “ITIL V3 Transición del Servicio” [5]. El concepto de Activo de Servicio, implícito en la V2, se hace explícito en V3.

El libro sobre estrategia del servicio [6] dedica varios apartados a definir que son los activos, centrándose en los activos de servicio, y dedicando uno a analizar la gestión financiera de los activos. Define activo como los recursos y capacidades utilizados por las organizaciones para crear valor en forma de bienes y servicios. Es la creación de valor lo que convierte un elemento en activo, un mismo elemento puede ser un activo para una organización y no para otra, por la incapacidad de esta última para crear valor con él.

Las organizaciones proporcionan a los clientes los bienes y servicios que estos demandan. Las organizaciones utilizan los activos para producir estos bienes y servicios creando valor en el proceso.

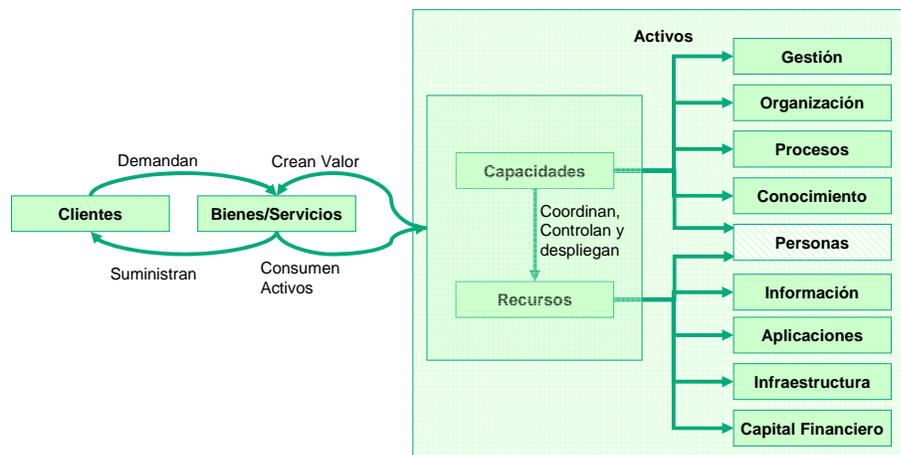


Figura 2. Papel de los activos en las organizaciones, según el libro de estrategia de Servicio

Los recursos son insumos directos para la producción. Son recursos el capital financiero, la infraestructura, las aplicaciones, la información y las personas. Las capacidades se utilizan para transformar los recursos, son capacidades la gestión, la organización, el conocimiento y las personas. Las personas son a la vez recursos y capacidades. Las capacidades representan la habilidad de las organizaciones para coordinar, controlar y desplegar los recursos con el objetivo de crear valor. Estas capacidades son intensivas en conocimiento y se encuentran embebidas en las personas, los procesos y los sistemas de las organizaciones. Es mucho más fácil adquirir recursos que capacidades.

Otro de los conceptos destacados en ITIL V3 es el concepto de ciclo de vida, aplicado al servicio y a los activos. El ciclo de vida es la secuencia de estados por los que pasa el activo o el servicio desde su adquisición o creación hasta su retiro o eliminación. Los activos hay que gestionarlos a lo largo de todo el ciclo de vida, desde que se realiza la petición del mismo, se compra o se construye, se prueba, se despliega para su operación, y llegado el momento se retira y elimina.

La gestión de activos abarca todo el ciclo de vida de los elementos, centrándose principalmente en aspectos financieros y los asociados con la regulación. Entre otros hay que gestionar el coste, la propiedad, la duración del servicio, el suministrador, las garantías y mantenimientos asociados,...

La gestión de la configuración, cuya misión es dar soporte a los procesos de la gestión del servicio, se centra principalmente en las etapas de despliegue y operación de los activos, centrándose en aspectos como el estado operativo, la versión, sub-CIs que pueda contener, su ubicación, el responsable y las relaciones con otros CIs.



Figura 3 Ciclo de Vida del Servicio

La base de datos de la gestión de la configuración sigue siendo la pieza clave para poder gestionar el servicio dentro de los parámetros de calidad y costes establecidos. La

CMDB da soporte a los procesos de gestión de incidencias, problemas, cambios, niveles de servicio,... ITIL V3 reconoce que la práctica en las organizaciones es la existencia de otros repositorios para la gestión de la configuración, además de la CMDB. Es práctica habitual la existencia de una librería de medios definitiva (DML), autónoma de la CMDB. Pero es la CMDB la que tiene que mantener las relaciones entre los CIs y estos repositorios externos, utilizando para ello artefactos de procesos.

Un repositorio de configuración clave en las organizaciones, para la ayuda a la toma de decisiones y el control de costes es la base de datos de activos, encargado del seguimiento y gestión de los activos a lo largo de todo el ciclo de vida.

Hay elementos que son a la vez CIs y activos. Todos los elementos que están en desarrollo y operación, sobre los que se realiza una gestión financiera y es necesario gestionarlos para la operación son a la vez activos y CIs. Existen otros elementos que son solo activos, como es el caso de los elementos que se han solicitado pero aún no se han recibido.

Otros son solo elementos de configuración. Es el caso, por ejemplo, de aquellos CIs que forman parte de otro CI, que son necesarios para el control del servicio y que por lo tanto están bajo el control de la CMDB, pero que se han adquirido dentro de un activo que lo contiene, y que desde el punto de vista financiero y de regulación no tienen existencia autónoma. Un ejemplo de esto puede ser el sistema operativo de un puesto de trabajo, que hay que gestionarlo para la gestión del servicio, pero que en su adquisición forma parte inseparable del puesto de trabajo y no se suele tener en cuenta desde el punto de vista de la gestión financiera. La decisión de si un elemento es activo, CI o ambos es parte del proceso de la gestión de la configuración y activos de servicio. Al conjunto de repositorios que soportan la gestión de configuración se denomina Sistema de Gestión de la Configuración (*Configuration Management System: CMS*) en ITIL V3.

Además de los elementos de configuración y los activos de servicio, existe otra información que es necesario gestionar para realizar una gestión eficiente del servicio. Es el caso de información como métricas sobre tiempo meteorológico, en aquellas organizaciones que este afecte a los procesos de la organización, el rendimiento de la organización, las capacidades de los proveedores, conocimientos del personal, necesidades de formación, cursos,... Toda esta información se integra en el sistema de gestión del

conocimiento (Service Knowledge Management System – SKMS), para realizar una gestión integral.

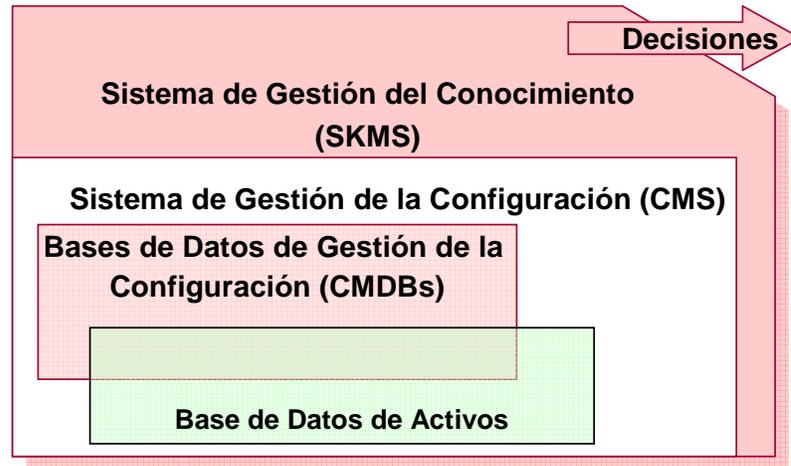


Figura 4. Relación entre los distintos repositorios del sistema de gestión del conocimiento

4. Gestión de la configuración y activos de servicio

El sistema de gestión de la configuración (CMS) da soporte a los distintos procesos y funciones que soportan la gestión de servicio. El CMS se mantiene actualizado mediante el proceso de Gestión de la configuración y Activos de Servicio (*Service Asset and Configuration Management: SACM*).

En aquellas organizaciones que mantienen la CMDB y la base de datos de activos como repositorios autónomos se pueden distinguir dos subprocesos claros, el subproceso de gestión de la configuración y el subproceso de la gestión de activos, aunque estrechamente relacionados.

Aunque ambos procesos se centran en elementos similares de la infraestructura de TI, los motivos para ello son diferentes. La gestión de activos realiza el seguimiento de los activos por razones financieras y legales a lo largo de todas las fases de su ciclo de vida, mientras que la gestión de configuraciones realiza el seguimiento de los CI y de las relaciones entre los CI para dar soporte a Gestión de incidencias, Gestión de problemas, Gestión de cambios y a otros procesos.

El objetivo de la gestión de configuración es mantener y proporcionar información precisa a todos los procesos que la necesiten sobre los elementos de configuración (CI) y sus relaciones en un modelo lógico, identificando todos los elementos de configuración en la infraestructura y en los servicios proporcionados a los clientes. Permite corregir las

discrepancias entre los registros de configuración y la configuración real de los distintos CIs, verificando la fidelidad de los registros de configuración.

Incluye el establecimiento de convenios de denominación para elementos de configuración y relaciones, el diseño, creación, relleno y actualización de la base de datos de gestión de configuración (CMDB), el soporte de auditorías de elementos de configuración, la identificación de interdependencias de elementos de configuración y sirve de enlace de cambios de elementos de configuración con RFC específicos. La definición de líneas base de configuración, para la comparación de esta con los elementos de configuración existentes es otra de sus funciones básicas.

El objetivo de la gestión de activos es maximizar el valor obtenido de los activos tecnológicos, y optimizar el coste de propiedad, dando soporte a la toma de decisiones de TI informadas, tanto en el ámbito estratégico como el táctico. Se reduce la exposición a los riesgos asociados con los activos de TI, garantizando el cumplimiento de los estándares legales, industriales y corporativos, y de los requisitos relacionados con los activos de TI. Incluye actividades como la gestión de licencias, asegurando la conformidad del uso del software con licencia del mismo, la administración de préstamo y mantenimiento de cada activo, la gestión de inventario, la asignación de activos disponibles para cumplir las solicitudes aprobadas, la activación de retiro de compra de activos caducados por otros nuevos y, en definitiva, el control del ciclo de vida financiero de activos.

5. Implementación de la gestión de los activos de servicio y la configuración

Para implementar un proceso de Gestión de la configuración y de Activos del Servicio (SACM) efectivo y eficaz hay que automatizar todo lo posible la alimentación y mantenimiento del Sistema de Gestión configuración (CMS). La automatización que es imprescindible en empresas medianas y grandes, puede hacerse igualmente en pequeñas organizaciones, haciendo uso de sencillas herramientas de inventario y de herramientas de descubrimiento automático de dispositivos y sus relaciones.

Los distintos Elementos de Configuración (CIs) y Activos de Servicio se incorporan al CMS por dos vías principales: por petición de algún proceso de gestión del servicio distinto del SACM y por una actividad de inventario planificada dentro del proceso SACM.

Durante la ejecución de algunas actividades en distintos procesos de gestión del servicio es necesario invocar al SACM para que realice un control de configuración que añada, modifique o elimine un CIs o un Activo de servicio. Por ejemplo en una petición de cambio para añadir un nuevo servidor, es necesario solicitar al proceso de SACM que añada este nuevo servidor al CMS. El SACM añadirá un nuevo CI con el nuevo servidor mediante una actividad de control de la configuración, e informará de la creación del mismo al solicitante del mismo. En una empresa con un grado de madurez en la gestión del servicio, esta debería ser la principal forma de alimentar el CMS. Si tenemos automatizado los distintos procesos de gestión del servicio con herramientas como las que proporciona la Plataforma de Tivoli para la Automatización de Procesos (Tivoli Process Automation Platform: TPAP), la alimentación del CMS se realizará automáticamente.

Por muy madura que se encuentre una organización, los procesos dependen en ultima instancia del grado de conocimiento y del compromiso de las distintas personas que intervienen en los mismos. Las herramientas de automatización de procesos facilitan el seguimiento de los procesos y reducen los errores en los mismos, pero no pueden impedir que se realicen actividades al margen de dichos procesos. Si es así la información contenida en el CMS no se encontrará actualizada, por ejemplo, porque alguien ha introducido un cambio en la configuración de una determinada aplicación sin seguir el procedimiento de gestión de cambios. Por ello es necesario incluir una actividad de inventario y auditoria en el proceso SACM.

Esta actividad de inventario, hay que realizarla al menos, en la carga inicial del CMS, pero conviene realizarla periódicamente. Para que este inventario sea efectivo conviene hacer uso de las herramientas de inventario automático y autodescubrimiento existentes. Hoy día la práctica totalidad de los CIs y Activos de Servicio, se encuentran conectados a la red IP y son por tanto susceptibles de ser inventariados de forma automática. A este descubrimiento automático se le puede complementar con información disponible en otras herramientas de software existentes en la organización.

Existen dos tipos de herramientas de inventario automático. Las basadas en la instalación de un agente en cada uno de los servidores y puestos de trabajo a inventariar. Un ejemplo de esta herramienta es el módulo de inventario del Tivoli Provisioning Manager. El otro tipo de herramientas de inventario automático son aquellas que no

necesitan instalar ningún programa en los elementos a inventariar, sino que interrogan a estos mediante APIs y comandos nativos disponibles en los dispositivos. Un ejemplo de herramienta sin agente es Tivoli Application Dependency Discovery Manager que además de descubrir los distintos servidores, elementos de red, puestos de trabajo y los programas y aplicaciones instalados en ellos, es capaz de descubrir como se relacionan los distintos CIs entre sí. Así es capaz de descubrir que se reciben peticiones web a través de un balanceador de carga, que a su vez reparte estas a uno o varios servidores Web, que a su vez están conectados una serie de servidores de aplicaciones que hacen peticiones a una base de datos o a un monitor transaccional en un Mainframe. La inclusión de estas relaciones entre CIs es uno de los elementos distintivos de una CMDB, en su función de soporte a los procesos de gestión de servicio de las tecnologías de la información.

Pese a cierto debate entre distintos fabricantes sobre las ventajas e inconveniente de uno y otro tipo de herramientas de inventariado automático, nosotros pensamos que ambas pueden formar parte del proceso de inventariado automático del CMS, y que cuanto más compleja es la organización más probable es que sean necesarias ambas.

Las herramientas que utilizan agente son capaces de identificar un mayor número de aplicativos y servir de soporte efectivo a procesos de auditoria que detecten la instalación de programas no autorizados o sin licencia en las organizaciones, e incluso son capaces de medir el uso de las distintas aplicaciones para la optimización de licencias adquiridas o para implementar programas de cobro por uso de la infraestructura. Sin embargo, para realizar su función es necesario instalar previamente el agente, con lo que dejan de ser útiles en los dispositivos conectados a la red en los que no se haya instalado el agente.

Por otra parte, las herramientas de descubrimiento sin agente, permiten hacer un *pooling* de la red, e identificar cualquier elemento conectado. Por el tráfico recibido y generado por el dispositivo es posible conocer que tipo de dispositivo es, si se trata de un *router*, de un servidor, si este utiliza el sistema operativo unix, linux, Si se dispone de credenciales de acceso al dispositivo, estas herramientas pueden obtener más información de configuración del dispositivo. Estas herramientas son capaces de detectar las relaciones entre los distintos CIs, analizando los patrones de tráfico y/o los ficheros de configuración de los distintos aplicativos, lo que les hace especialmente valiosas para automatizar el descubrimiento de las relaciones necesarias en la CMDB y para la realización del análisis

del impacto que las distintas actividades y decisiones tienen sobre los servicios proporcionados.

Según Gartner [7] las herramientas de CMDB deben de tener las capacidades de reconciliación, federación, sincronización y visualización. Aprovechando estas capacidades, proponemos una aproximación *bottom-up* de construcción del CMS, partiendo del descubriendo automático de la infraestructura, su configuración y sus relaciones para crear el conjunto de CIs autorizados necesarios para dar soporte a los procesos de gestión del servicio.

La aproximación que proponemos para automatizar el inventario y reconciliación del CMS puede observarse esquemáticamente en la siguiente figura:

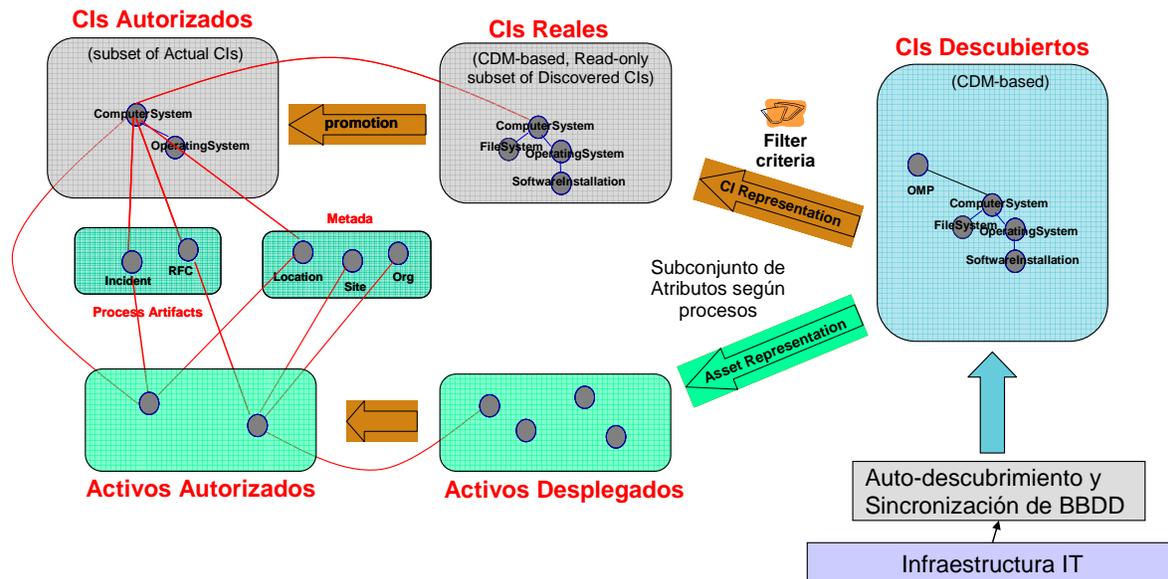


Figura 5. Automatización: proceso de autodescubrimiento, sincronización y reconciliación

Partimos de una o varias herramientas de descubrimiento automático de la infraestructura como las descritas anteriormente para identificar los distintos CIs que hay conectados a la red. Todos estos CIs con sus atributos y relaciones se cargan en un repositorio de datos con los CIs descubiertos. Como el descubrimiento de la infraestructura se hace de forma periódica y continua, al cargar cada CI en este repositorio, hay que comprobar si existe el CI en el repositorio, para no crear CIs duplicados. Si ya existe el CI, cargaremos solo aquellos atributos que no estuvieran en el repositorio, o que hayan

cambiado desde el último descubrimiento realizado. En el caso de que haya cambios anotaremos los mismos, para tener un inventario de cambios realizados en los CIs.

A esta carga controlada de CIs la denominamos reconciliación y se puede automatizar mediante herramientas de autodescubrimiento con reglas de correlación avanzada [8], [9]

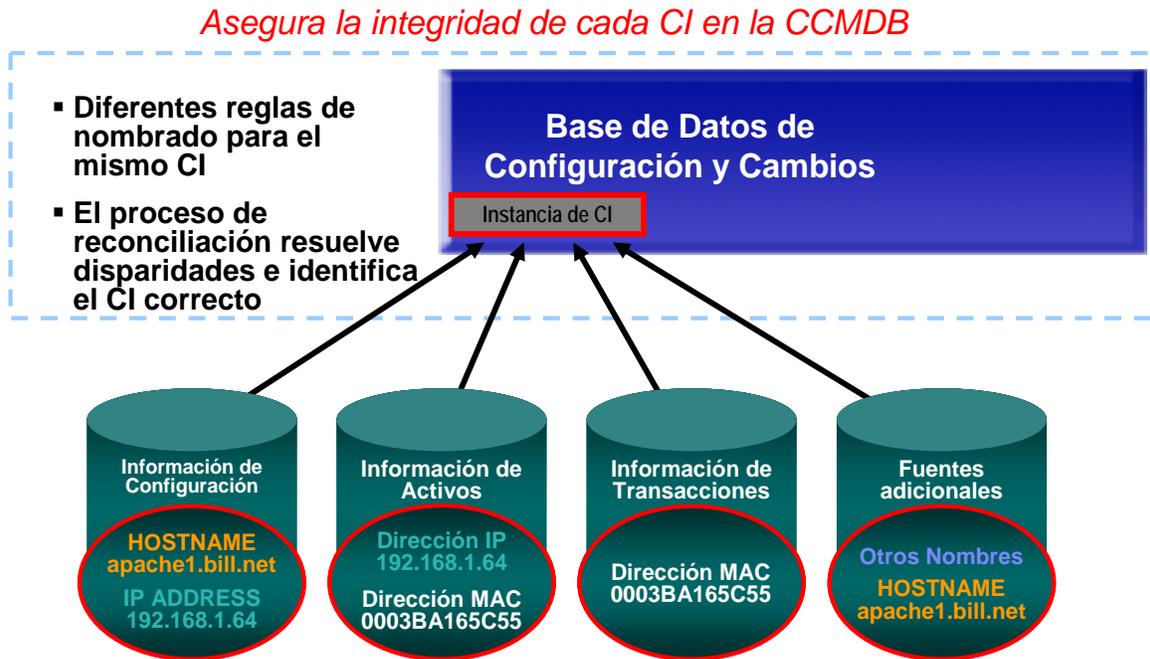


Figura 6. Asegurar la integridad de cada CI en la CCMDB

En el caso de disponer de información de configuración residente en distintas herramientas, se reconciliará la información de los CIs y atributos de cada herramienta con los existentes en el repositorio de CIs descubiertos. Se asegura así que independientemente de cómo se denominen los CIs en cada herramienta, estos no se duplican en el repositorio de CIs descubiertos. Así es posible tener centralizada en un único punto de forma automática toda la información de configuración de la organización.

Este proceso de reconciliación se puede realizar no solo con herramientas de descubrimiento automático, sino que se puede realizar contra cualquier repositorio existente en la organización para otros propósitos. Por ejemplo, podemos cargar información procedente del sistema de contabilidad, de repositorios de proveedores externos, etc.

En muchos casos no hay que duplicar la información. Es posible, mediante las tecnologías de federación de datos, crear una relación virtual entre el repositorio de CIs

descubiertos y otros repositorios de información, de forma que se pueda consultar la información en tiempo real desde el CMS de cualquier atributo, pero no es necesario duplicar ingentes cantidades de información con las ineficacias que ello generaría.

Gracias a la automatización, este repositorio de CIs descubiertos es una fuente de información fiable de gran ayuda a los distintos procesos y actividades de la organización. Así un técnico de comunicaciones puede utilizar esta información para ver si la configuración de un determinado *router* es conforme a la configuración estándar, línea base, de comunicaciones en la organización. Un técnico de servidores de aplicaciones, puede ver las diferencias de configuración entre dos clones de un servidor. Un técnico que está analizando un problema puede comprobar si se ha realizado algún cambio de configuración en los elementos que soportan el servicio en cuestión, comparando los parámetros de configuración actuales, con los que había antes de que apareciera el problema.

Aunque podríamos utilizar este repositorio como almacén para la gestión de la configuración o CMDB, y como repositorio para la gestión de Activos, no es práctico trabajar con todos los CIs y los atributos de configuración de cada uno de los CIs descubiertos. Por ejemplo, con las herramientas de autodescubrimiento podemos conocer los programas que hay en un servidor y la configuración de cada programa, los sistemas de ficheros que hay, los discos que los soportan y mucha más información. Para dar soporte a un proceso, como por ejemplo, el proceso de gestión de cambios, podemos decidir que es suficiente trabajar al nivel de servidor y que tratar como CIs los distintos sistemas de ficheros, crea una complejidad innecesaria para la estructura de la organización.

Con el objeto de simplificar el tratamiento de procesos, proponemos filtrar los CIs descubiertos, tanto en número de CIs, como en atributos, para crear un conjunto manejable de información que denominamos CIs reales. Este proceso de filtrado se realiza automáticamente y se define durante la planificación de la gestión de configuración.

La CMDB contiene el conjunto de CIs autorizados, que han sido creados mediante los procesos formales, como la gestión de cambios. En la fase de verificación del proceso de SACM se procede a comparar el conjunto de CIs reales con los autorizados, identificando las discrepancias y promoviendo las RFCs necesarias para que la CMDB mantenga una información fidedigna en todo momento.

Al igual que se filtran los CIs descubiertos para crear un conjunto manejable de CIs reales, se puede aplicar otro filtro para configurar un conjunto de activos desplegados, que en la fase de auditoria del subproceso de gestión de activos se promueven mediante RFCs a activos autorizados, que son aquellos activos contenidos en la base de activos procedentes del proceso formal de gestión de activos.

Igual que se ha realizado con la CMDB y la Base de datos de Activos, mediante este mecanismo de filtrado es posible mantener actualizado los distintos repositorios que conforman el CMS.

6. Suite de Tivoli para la gestión de servicio TI

Esta filosofía de implementación es fácilmente implementable mediante la suite de Tivoli para la Gestión de Servicios TI. La suite de Tivoli para la Gestión del Servicio TI es un conjunto de herramientas flexibles, sobre una plataforma SOA estándar, que permiten adoptar las mejores prácticas de ITIL V3 de una forma eficiente. Sobre una misma plataforma se implementan los distintos procesos, siendo posible implementar la gestión de activos y la gestión de elementos de configuración, tal como se ha descrito anteriormente. La configuración de la plataforma en base a roles permite asignar los distintos procesos a las mismas o a distintas personas, proporcionándose una plataforma que es eficaz en pequeñas compañías y que escala en empresas con decenas de miles de empleados. Los módulos para implementar los procesos descritos son *Tivoli Asset Manager for TI* para desplegar la gestión de activos, y *Tivoli Application Dependency Manager* para implementar la gestión de configuración.

7. Conclusión

En ITIL V3, se reconoce la importancia de los activos de servicio para proporcionar valor a las organizaciones, diferenciándolos de los elementos de configuración. El proceso de gestión de configuración y activos de servicio permite mantener actualizada la información de CIs y de activos. Hemos abordado una forma de mantener actualizada de forma automática esta información, desglosando la información entre la CMDB y la Base de Datos de activos, mediante el uso de herramientas automáticas de autodescubrimiento, reconciliación y federación de datos. Esta forma de implementación puede extenderse para

definir otros repositorios específicos de información con los que se necesite completar el sistema de gestión de la configuración.

Existen otras aproximaciones para la implementación del CMS, basadas en una aproximación *top-down*, donde se decide primero los atributos que necesitan los distintos procesos y luego se identifica la fuente o forma de alimentar el dato. Creemos que nuestra aproximación *bottom-up*, de descubrir mediante herramientas de autodescubrimiento la información de configuración de los CIs y luego seleccionar de entre ellos, atributos que van a ser necesarios para los distintos procesos, añadiendo algún atributo si fuera necesario, permite una implementación más rápida y eficaz de los procesos de gestión del servicio, y permite demostrar más rápidamente a las organizaciones el valor de implementar las mejores prácticas de ITIL.

Referencias

- [1] IBM, *A Management System for the Information Business*, IBM, 1980
- [2] M. Ernest and J. M. Nisavic, *Adding Value to the IT Organization with the Component Business Model*, IBM Systems Journal 46, No. 3, 2007, pp. 387–403
- [3] IBM Corporation, IBM Tivoli Unified Process, IBM Corporation, <http://www.ibm.com/software/tivoli/governance/servicemanagement/itup/tool.html>, 2008
- [4] Berkhout, M., Harrow, R., Johnson, B., Lacy, S., Lloyd, V., Page, D., van Goethem, M., van den Bent, H. y Welter, G, *ITIL V2 Service Support*, The Office of Government Commerce under licence from the Controller of Her Majesty's Stationery Office, 2000.
- [5] Taylor, S. , Shirley Lacy, Ivor MacFarlane, *ITIL Service Transition*, Published by TSO (The Stationery Office) for the Office of Government Commerce under licence from the Controller of Her Majesty's Stationery Office, 2007
- [6] Iqbal, M. y Nieves, M., *ITIL Service Strategy*, Published by TSO (The Stationery Office) for the Office of Government Commerce under licence from the Controller of Her Majesty's Stationery Office, 2007
- [7] Colville, R. J., *CMDB or Configuration Database: Know the Difference*, RAS Core Research Note G00137125, Gartner, Inc., Stamford, CT 06904, marzo, 2006, <http://mediaproducts.gartner.com/gc/reprints/ibm/external/article5/article5.html>.

- [8] Ward, C., Aggarwal, V., Bucu, M., Olsson, E. y Weinberger S., “Integrated change and configuration management”, *IBM Systems Journal*, vol. 46, nº 3, pp. 459-478, 2007.
- [9] Madduri, H., Shi, S. S. B., Baker, R., Ayachitula, N., Shwartz, L., Surendra, M., Corley, C., Benantar, M. y Patel, S., “A configuration management database architecture in support of IBM Service Management”, *IBM Systems Journal*, vol. 46, nº 3, pp. 441-457, 2007.

Reseña sobre las V Jornadas de Testeo de Software (JTS'08)

Tanja E.J. Vos
Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
tanja@iti.upv.es

Más de 150 asistentes acudieron a la V edición de las “Jornadas sobre Testeo de Software” (JTS2008) organizadas por el Instituto Tecnológico de Informática (ITI) y celebradas durante los días 2, 3 y 4 de abril de 2008 en Valencia, dentro del campus de la Universidad Politécnica de Valencia (UPV). El evento JTS se inició en el año 2004 y ya se ha consolidado como uno de los principales referentes nacionales en el área del testeo de software. Esta quinta edición contó con la participación de expertos internacionales en el área de software testing como Lee Copeland (SQE, Estados Unidos), Julian Harty (Google, Reino Unido), Bart Knaack (Logica, Países Bajos), Mieke Gevers (AQIS, Bélgica), Petra Heck (Universidad de Eindhoven, Países Bajos) y Graham Thomas (BadgerSoft, Reino Unido). Además, diversos expertos nacionales contaron sus experiencias en el campo del testeo de software.

El evento se inició el miércoles, 2 de abril, con un seminario complementario de Lee Copeland titulado “Black Box Test Design Techniques”, especialmente orientado a algunas técnicas para testeo de tipo “caja negra”). Durante la jornada del 3 de abril, después de la bienvenida y presentación a cargo del director científico del ITI, tuvieron lugar las primeras ponencias en inglés (con traducción simultánea): “Elegir lo mejor del testeo planificado y ágil” (Lee Copeland); “Testeo de rendimiento (una metodología de 10 pasos)” (Mieke Gevers); “Ingeniería de Testeo en Google” (Julian Harty); “Todo lo hay que saber sobre testeo de software se aprende en el jardín de infancia” (Lee Copeland).

Tras el almuerzo se realizaron las charlas en español: “Estudios de rendimiento y monitorización de aplicaciones Web” (Salvador I. Folgado Bellido); “Testeo en mi empresa... ¿Cómo empiezo?” (Tanja Vos); “Aspectos destacados del Performance Testing” (Fernando Rodríguez / Xavier Cuadrado); “SSTQB (Spanish Software Testing Qualification Board)” (José Díaz); “REPRIS (REd para las PRuebas en la Ingeniería del Software)” (Luis Fernandez Sanz); “Cómo asegurar la fiabilidad de los Sistemas de

Información a través de las pruebas” (Jean-Jacques Messina); “INTECO. Diagnóstico de la calidad del software en España” (Antonio Sepúlveda Carrer). La jornada terminó con una mesa redonda en la que los ponentes del día respondieron a diversas cuestiones planteadas por los asistentes.



El 4 de abril siguió la misma estructura que la jornada anterior, con las ponencias en inglés durante la sesión matutina: “El principio de los plátanos para el testeo de software” (Lee Copeland); “¿Listos para el testeo? Cómo crear y validar requerimientos de usuario” (Petra Heck); “Definiendo e implementando un enfoque de testeo en toda la empresa” (Graham Thomas); “Diez habilidades esenciales para el Testeo de Software Estructurado” (Bart Knaack). En la última sesión, los ponentes realizaron las siguientes exposiciones en español: “Un framework de testeo: enfoque básico para los directivos” (Javier Fernández-Pello Alvargonzález); “Gestión de pruebas y validación de sistemas con Telelogic DOORS” (Antonio Rodríguez); “¿Por qué TestLink? Un caso de éxito” (Fernando David Vicente Márquez); “Probando desde el principio (Antonio Fuentes Soto)”. Después de una nueva mesa redonda, las jornadas se clausuraron con un vino de honor.

El evento incluyó también expositores de las empresas colaboradoras (Bull, TestABil y Steria) en las que los asistentes pudieron conocer, de primera mano, sus herramientas de testeo y otros productos.

Las presentaciones utilizadas por los ponentes están disponibles en la web oficial del evento (<http://www.iti.upv.es/JTS2008>); próximamente se publicarán los proceedings de JTS2008 con la recopilación de artículos de los ponentes.

Reseña sobre el III Congreso Interacadémico 2008-itSMF España

Antonio Folgueras
Departamento de Informática
Universidad Carlos III de Madrid
afolguer@inf.uc3m.es

El III Congreso Interacadémico 2008-itSMF España se celebró a lo largo del día 13 de octubre de 2008 en la Universidad Carlos III de Madrid. Tras la publicación de ITIL v3, se analizó como las mejoras en el ciclo de vida de las tecnologías reportan valor a las organizaciones. En el encuentro se dieron cita un gran número de profesionales, investigadores y estudiantes interesados en las mejores prácticas de la Gestión de Servicio de las TI. En el presente Congreso Interacadémico, también participaron las Facultades y Escuelas de Informática de: Universidad de Alcalá, Universidad Nacional de Educación a Distancia, Universidad Antonio de Nebrija, Universidad Politécnica de Madrid, y Universidad Rey Juan Carlos, así como la Escuela de Ingeniería de Tele-comunicaciones de la Universidad Pompeu Fabra.

Los objetivos del congreso eran ambiciosos pues abarcaban un amplio espectro desde ser una catapulta de la innovación a seguir potenciando la difusión del conocimiento sobre gestión de servicios TI. Aparte dicho congreso permitió a los estudiantes de últimos cursos de ingeniería tecnológica el tomar contacto con el mundo empresarial e investigador. La asistencia al congreso fue numerosa superando los trescientos participantes y excediéndose en algunos momentos la capacidad de los recintos.



El congreso fue inaugurado por D. Mark Gemmell vicepresidente de itSMF España y la Dra. Belén Ruiz Mezcua Vicerrectora Adjunta de Investigación para el Parque Científico Universidad Carlos III de Madrid. A lo largo de sus 21 ponencias en tres sesiones en paralelo y una mesa redonda se debatieron temas relativos a gestión del servicio y gobierno de las TI mezclando las visiones profesionales y académicas. En el ecuador del congreso, para relajar las mentes, se contó con un ameno show producido por Hydra Producciones.

El cierre del congreso y punto álgido del evento se materializó en una mesa redonda en donde se contó con un plantel compuesto por reconocidas personalidades relacionadas con el mundo de los estándares TI: D. Carlos Manuel Fernández (AE-NOR), Dña. Ana M^a Rodríguez de Viguri (AETIC), D. Manuel Monterrubio (ALI), D. Francisco Antón Vique (ASTIC) y D. Luis Fernández Sanz (ATI). En dicha mesa redonda moderada por D. Marlon Molina de New Horizons, se discutió el papel de los estándares y mejores prácticas en el papel del ingeniero tecnológico y el importante papel que en la adopción de estándares juegan las Administraciones Públicas. A lo largo de la mesa redonda también se dió fe de la escasez de encontrar profesionales tecnológicos y la importancia de asociarse y ser parte activa de las asociaciones que representan un sector que mueve un porcentaje muy significativo del PIB de España. Las ponencias pueden bajarse desde www.itsmf.es y también en formato video.

Un sociólogo estudia la producción de software: ¿qué puede aportar su mirada al estudio de la evolución del trabajo de los informáticos?

Juan José Castillo Alonso

Catedrático de Sociología

Director del Grupo de Investigación 'Charles Babbage' en Ciencias Sociales del Trabajo

<http://www.ucm.es/info/charlesb/>

Facultad de Ciencias Políticas y Sociología, Universidad Complutense de Madrid

Campus de Somosaguas, 28223 Pozuelo de Alarcón (Madrid)

jjcastillo@cps.ucm.es

El estudio de campo

El estudio de las tendencias de la evolución de la producción de software ha venido analizándose en la literatura sociológica desde hace más de tres décadas, con un énfasis especial en las formas que adopta la organización del trabajo, la división de la inteligencia aplicada a la producción, y la reorganización empresarial. Y ello con algunos rasgos característicos que identifican, a mi juicio, la posible singularidad de nuestra aportación.

En primer lugar, se trata de llevar a cabo un estudio de terreno, teóricamente orientado, capaz de separar lo que debe ser de lo que es. Dicho en los términos acuñados por la antropotecnología de Alain Wisner, se trata de mostrar no sólo el trabajo y la organización del mismo teórica o prescrita, sino la actividad y la organización real.

En la investigación sobre el desarrollo de software, la construcción de programas, el ciclo de vida, que comprende desde la toma de requisitos del “cliente”, el diseño, la arquitectura, el análisis funcional, las pruebas parciales y de conjunto, la prueba, la aplicación y el mantenimiento, es especialmente indicado esta manera de mirar.

Porque, en numerosas ocasiones, como han analizado con brillantez los investigadores daneses Hansen, Rose y Tjornehoj (2004), para un conjunto de 322

investigaciones sobre métodos de mejora de los procesos de desarrollo de software, lo que predomina es más la prescripción, que la descripción o la reflexión. Su conclusión no puede ser más esclarecedora: la inmensa mayoría de los artículos u obras analizadas dicen como deben ser las cosas, pero no necesariamente como son. Incluso, para destacar su argumento, en la versión en documento de trabajo, los autores juegan con los tipos de imprenta para darle un tamaño gigante a la “prescription” frente a la “description”, reduciendo a un tamaño minúsculo la “reflection”.

Por ejemplo, estudiar los “problemas reales de los equipos virtuales”, es una condición indispensable para poder dar cuenta de los cambios reales que están teniendo lugar en estos procesos productivos y en la sociedad. Aunque ello haga, naturalmente, más complejo tanto el abordaje teórico, como los temas abordados.

Esta voluntad de reconstruir “las situaciones reales de trabajo”, lo realmente existente, es una necesaria marca epistemológica en el caso del software, puesto que es más que habitual, no sólo la generalización, con escaso fundamento, respecto al propio ‘sector’, sino su transferencia a los cambios globales de la sociedad como un todo. Y si no, veáse lo que dicen dos catedráticos, uno de Harvard y otro de Berlín, en un elegante prologo al libro éxito de ventas *Secrets of software success* (1999): “en el centro de este libro, sin embargo [en relación a otros problemas de gestión, según el sector: el software empaquetado, las soluciones empresariales y los servicios profesionales], está un muy diferente enfoque que demanda esta industria a la gestión de recursos humanos. Las jerarquías rígidas de la era industrial; los caminos de carreras largas, y así sucesivamente, no funcionan aquí. Este es, genuinamente, un mundo diferente”.

Situación en el entorno

En segundo lugar, para poder situar el estudio de los procesos actuales de trabajo en el desarrollo del software, de la construcción de programas informáticos, utilizamos una metodología de trabajo que trata de colocar cada proceso productivo en su contexto más amplio, en la misma línea de abordaje que lo que ha llamado Michael Burawoy (1998), con gran acierto, “the extended case method”. Colocando así los estudios empíricos, artesanos, minuciosos y detallados en un marco explicativo que les da sentido y profundidad.

Nuestro punto de partida, se plasma en un marco teórico, fundado en muy distintas investigaciones empíricas, e incluye la consideración de las policy options, las opciones de políticas razonables y razonadas, de crear entornos donde los círculos virtuosos de sinergias y recursos públicos y privados, puedan dar origen a distritos, clusters, desarrollos locales endógenos, que permitan garantizar una opción de desarrollo tanto personal como institucional y regional sostenible, y que transite por la vía alta del desarrollo económico y social. Punto de partida que entronca con el mainstream, el marco de análisis actual en nuestra comunidad científica, que se apoya en trabajos muy semejantes a los que hemos desarrollado en nuestro equipo: como la división del trabajo entre empresas (Grimaldi y Torrisi, 2001); los problemas de gobierno estratégico de las redes de empresas, (Gereffi et alii, 2005; Sturgeon, 2004); la evolución de la división del trabajo (Cappelli, 2001; Cusumano, 1992; Beirne, Ramsay y Panteli, 1998), etc.

Esta literatura fundamenta el hecho de que aquello que formaba el núcleo central de la “nueva división internacional del trabajo” (Fröbel et alii, 1980), basado en la externalización de trabajo descualificado, se dobla, hoy en día con la posibilidad, y la realidad, desde luego, en este sector de la producción de software, de la externalización de trabajo cualificado, de trabajo inmaterial, de tareas que antes se consideraban sólo realizables en los países centrales.

El ‘sector’ de la producción de software es, en este sentido, un terreno especialmente adecuado para analizar el contexto, las fuerzas que lo mueven, las transformaciones y las consecuencias para el trabajo, que están, según muestra la investigación social, mudando cada día tanto en la conformación de las empresas, como en la vinculación entre ellas. Para ello, nuestro enfoque, basado en la reconstrucción de los procesos completos de producción, es especialmente esclarecedor.

Perfil profesional



Juan José Castillo es catedrático de Sociología en la Universidad Complutense de Madrid. Es autor de “El trabajo fluido en la sociedad de la información. Organización y división del trabajo en las fábricas de software” (Madrid y Buenos Aires, Editorial Miño y Dávila, 2007) y co-director de la revista Sociología del Trabajo (Editorial Siglo XXI). Una panorámica general de los trabajos de investigación y publicaciones del Grupo de Investigación de la Universidad Complutense que dirige, puede verse en la página www.ucm.es/info/charlesb/. El libro que mejor recoge una síntesis de la primera fase de los trabajos del proyecto de investigación “El trabajo invisible en España”, es “El trabajo recobrado. Una evaluación del trabajo realmente existente en España” (Madrid y Buenos Aires, Miño y Dávila , 2005). La tercera fase de ese proyecto acaba de ser concedida por el Ministerio de Ciencia y Tecnología para los años 2008 -2011, “Nuevos Modelos de Vida y Trabajo en la Sociedad de la Información. El caso de las grandes periferias metropolitanas. (TRAVIDA)”.